1 Claim 1 (currently amended): A computer-implemented method of programmatically building

2 queries, comprising steps of:

3   programmatically building a query user interface to query a content source, wherein the

4 query user interface comprises a plurality of query parameters, each query parameter comprising a

5 unique query parameter name, a query qualifier, and a query parameter value, further comprising:

6   dynamically identifying the content source to be queried;

7   programmatically determining a plurality of content values specified in the

8 dynamically-identified content source;

9   programmatically determining, based on the specified content values, a plurality of

10 content types corresponding thereto;

11 ~~programmatically identifying, for a content source, at least one element thereof by~~

12 ~~programmatically obtaining one or more tag names from a markup language document used for~~

13 ~~rendering a view of the content source;~~

14   using at least one of the ~~programmatically-obtained~~ programmatically-determined

15 ~~tag names~~ content types to consult a lookup component to obtain at least [[one]] two ~~candidate~~

16 query parameter [[name]] names usable to query ~~for querying~~ the content source;

17   programmatically identifying, for each of the obtained query parameter names, at

18 least one query qualifier corresponding thereto, each query qualifier usable in determining a match

19 when comparing selected ones of the content values to that query parameter name;

20   programmatically identifying, for at least one of the obtained query parameter

21 names, at least one value usable therewith as a query parameter value;

22   programmatically building the plurality of query parameters by associating, with

23 each of the obtained query parameter names, the programmatically-identified at least one query

24 qualifier corresponding thereto and the programmatically-identified at least one value usable

25 therewith as a query parameter value, if any; and

26   displaying on the query user interface, for each of the programmatically-built query

27 parameters, the obtained ~~candidate~~ query parameter name, a first selector usable to select one of

28 the at least one query qualifiers corresponding thereto, and a second selector usable to select at

29 least one of the at least one values usable therewith, if any, or for providing at least one user-

30 entered value usable therewith; and ~~on a user interface display configured~~

31 enabling a user to build a query command~~, responsive to selection by a user of at least one~~

32 ~~of the displayed candidate query parameter name or names,~~ to query the content source by using,

33 for each of at least one of the displayed query parameter names, the first selector to select one of

34 the associated query qualifiers and using the second selector to select at least one of: (1) at least

35 one of the associated values, if any, or (2) at least one user-entered value.

Claim 2 (canceled)

1 Claim 3 (previously presented): The method according to Claim 1, wherein the using step further

2 comprises using information regarding the user when consulting the lookup component.

1 Claim 4 (currently amended): The method according to Claim 1, further comprising the step of:

2   programmatically identifying at least one query extension parameter [[name]] for the query

3   command, responsive to a request from the user to extend the ~~query command~~ <u>display on the</u>

4   <u>query user interface, further comprising, for each of the at least one query extension parameters:</u>

5   <u>using at least one of the obtained query parameter names to obtain a related query</u>

6   <u>parameter name;</u>

7   <u>programmatically identifying at least one query qualifier corresponding to the</u>

8   <u>obtained related query parameter name, each query qualifier usable in determining a match when</u>

9   <u>comparing selected ones of the content values to the obtained related query parameter name; and</u>

10   <u>programmatically building the query extension parameter by associating, with the</u>

11   <u>obtained related query parameter name, the programmatically-identified at least one query</u>

12   <u>qualifier corresponding thereto</u>; and

13   wherein the displaying step further comprises also displaying each of the at least one

14   programmatically-identified query extension ~~parameter name or names~~ <u>parameters</u> as additional

15   ones of the ~~candidate~~ <u>programmatically-built</u> query ~~parameter names~~ <u>parameters</u>.

Claim 5 (canceled)

1   Claim 6 (previously presented): The method according to Claim 1, wherein the using step further

2   comprises using information regarding the content source when consulting the lookup component.

1   Claim 7 (previously presented): The method according to Claim 3, wherein the information

2   regarding the user comprises at least one of: a role of the user, preferences of the user, a device

3   used by the user, or an identification of the user.

Claims 8 - 23 (canceled)

1   Claim 24 (new): A system configured to programmatically build queries, comprising:

2   means for programmatically building a query user interface to query a content source,

3   wherein the query user interface comprises a plurality of query parameters, each query parameter

4   comprising a unique query parameter name, a query qualifier, and a query parameter value,

5   further comprising:

6   means for dynamically identifying the content source to be queried;

7   means for programmatically determining a plurality of content values specified in

8   the dynamically-identified content source;

9   means for programmatically determining, based on the specified content values, a

10   plurality of content types corresponding thereto;

11   means for using at least one of the programmatically-determined content types to

12   consult a lookup component to obtain at least two query parameter names usable to query the

13   content source;

14   means for programmatically identifying, for each of the obtained query parameter

15   names, at least one query qualifier corresponding thereto, each query qualifier usable in

16   determining a match when comparing selected ones of the content values to that query parameter

17   name;

18   means for programmatically identifying, for at least one of the obtained query

19   parameter names, at least one value usable therewith as a query parameter value;

20   means for programmatically building the plurality of query parameters by

21   associating, with each of the obtained query parameter names, the programmatically-identified at

22   least one query qualifier corresponding thereto and the programmatically-identified at least one

23   value usable therewith as a query parameter value, if any; and

24   means for displaying on the query user interface, for each of the programmatically-

25   built query parameters, the obtained query parameter name, a first selector usable to select one of

26   the at least one query qualifiers corresponding thereto, and a second selector usable to select at

27   least one of the at least one values usable therewith, if any, or for providing at least one user-

28   entered value usable therewith; and

29   means for enabling a user to build a query command to query the content source by using,

30   for each of at least one of the displayed query parameter names, the first selector to select one of

31   the associated query qualifiers and using the second selector to select at least one of: (1) at least

32   one of the associated values, if any, or (2) at least one user-entered value.

1   Claim 25 (new): A computer program product configured to programmatically build queries, the

2   computer program product embodied on one or more computer-readable storage media and

3   comprising:

4   computer-readable program code for programmatically building a query user interface to

5   query a content source, wherein the query user interface comprises a plurality of query

6   parameters, each query parameter comprising a unique query parameter name, a query qualifier,

7   and a query parameter value, further comprising:

8   computer-readable program code for dynamically identifying the content source to

9 be queried;

10 computer-readable program code for programmatically determining a plurality of

11 content values specified in the dynamically-identified content source;

12 programmatically determining, based on the specified content values, a plurality of

13 content types corresponding thereto;

14 computer-readable program code for using at least one of the programmatically-

15 determined content types to consult a lookup component to obtain at least two query parameter

16 names usable to query the content source;

17 computer-readable program code for programmatically identifying, for each of the

18 obtained query parameter names, at least one query qualifier corresponding thereto, each query

19 qualifier usable in determining a match when comparing selected ones of the content values to

20 that query parameter name;

21 computer-readable program code for programmatically identifying, for at least one

22 of the obtained query parameter names, at least one value usable therewith as a query parameter

23 value;

24 computer-readable program code for programmatically building the plurality of

25 query parameters by associating, with each of the obtained query parameter names, the

26 programmatically-identified at least one query qualifier corresponding thereto and the

27 programmatically-identified at least one value usable therewith as a query parameter value, if any;

28 and

29 computer-readable program code for displaying on the query user interface, for

30 each of the programmatically-built query parameters, the obtained query parameter name, a first

31 selector usable to select one of the at least one query qualifiers corresponding thereto, and a

32 second selector usable to select at least one of the at least one values usable therewith, if any, or

33 for providing at least one user-entered value usable therewith; and

34   computer-readable program code for enabling a user to build a query command to query

35 the content source by using, for each of at least one of the displayed query parameter names, the

36 first selector to select one of the associated query qualifiers and using the second selector to select

37 at least one of: (1) at least one of the associated values, if any, or (2) at least one user-entered

38 value.